**РУСЕНСКИ УНИВЕРСИТЕТ „АНГЕЛ КЪНЧЕВ“**

**КУРСОВА РАБОТА**

ПО

СОФТУЕРНО ИНЖЕНЕРСТВО

Студент: **Бейтула Хамди Реджеб**

Фак. Номер: **103217**

Група: **26А**

Специалност: **КСТ**

Дата: Проверил:

# Съдържание

[1. Задание 3](#_Toc344907667)

[2. Диаграми 4](#_Toc344907668)

[2.1 Delphi диаграма 4](#_Toc344907671)

[2.2 Use Case 5](#_Toc344907672)

[2.3 Sequence 6](#_Toc344907673)

[2.4 Activity 7](#_Toc344907674)

[2.5 Class 8](#_Toc344907675)

[3. Работа с програмата 9](#_Toc344907676)

[4. Особености на програмата 13](#_Toc344907677)

[5. Код на програмата 13](#_Toc344907678)

[5.1 Unit1.pas 13](#_Toc344907679)

[5.2 Unit2.pas 16](#_Toc344907680)

[5.3 Unit3.pas 20](#_Toc344907681)

# Задание

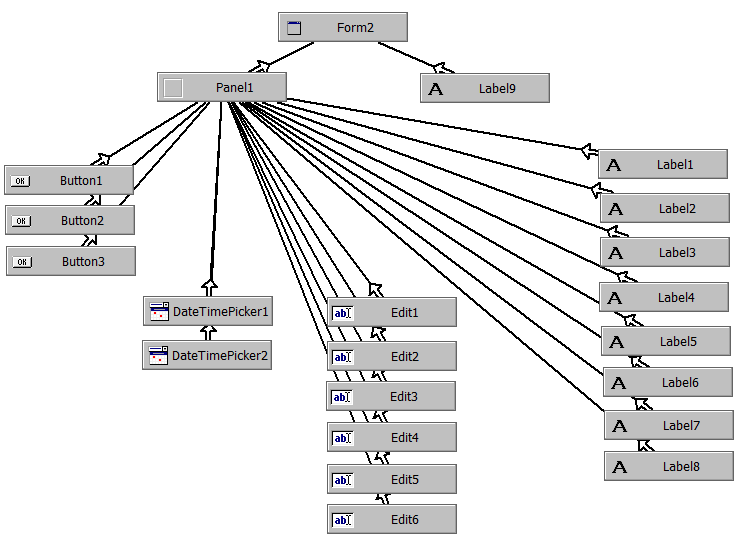
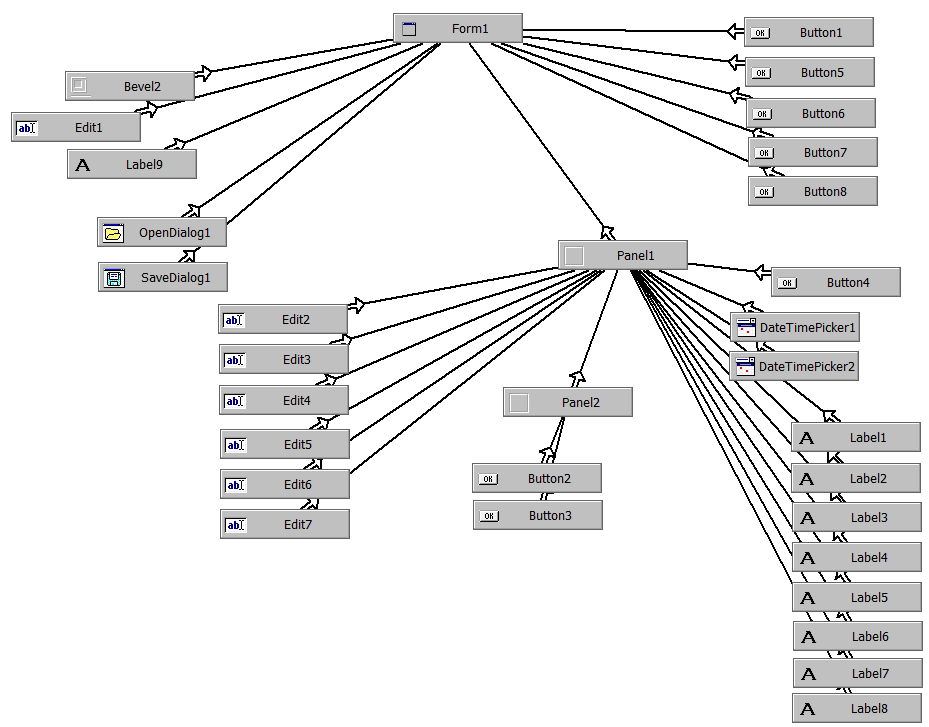
**Да се проектира и реализира Windows приложение за обслужване на индексно-последователен файл.** Инструменталното средство да обслужва следната структура от данни: име, презиме и фамилия на студента, дата на раждане, дата на записване, **факултетен номер,** група, среден успех и да предоставя на потребителя следните функционални възможности:

* Добавяне, редактиране, изтриване на запис;
* Търсене по индекс;
* Логически контрол на ключа и всички останали полета;
* Създаване на нов файл, четене от файл и запис във файл;
* Генериране на отчет с подадените команди, дата и време за начало и край на сеанса.

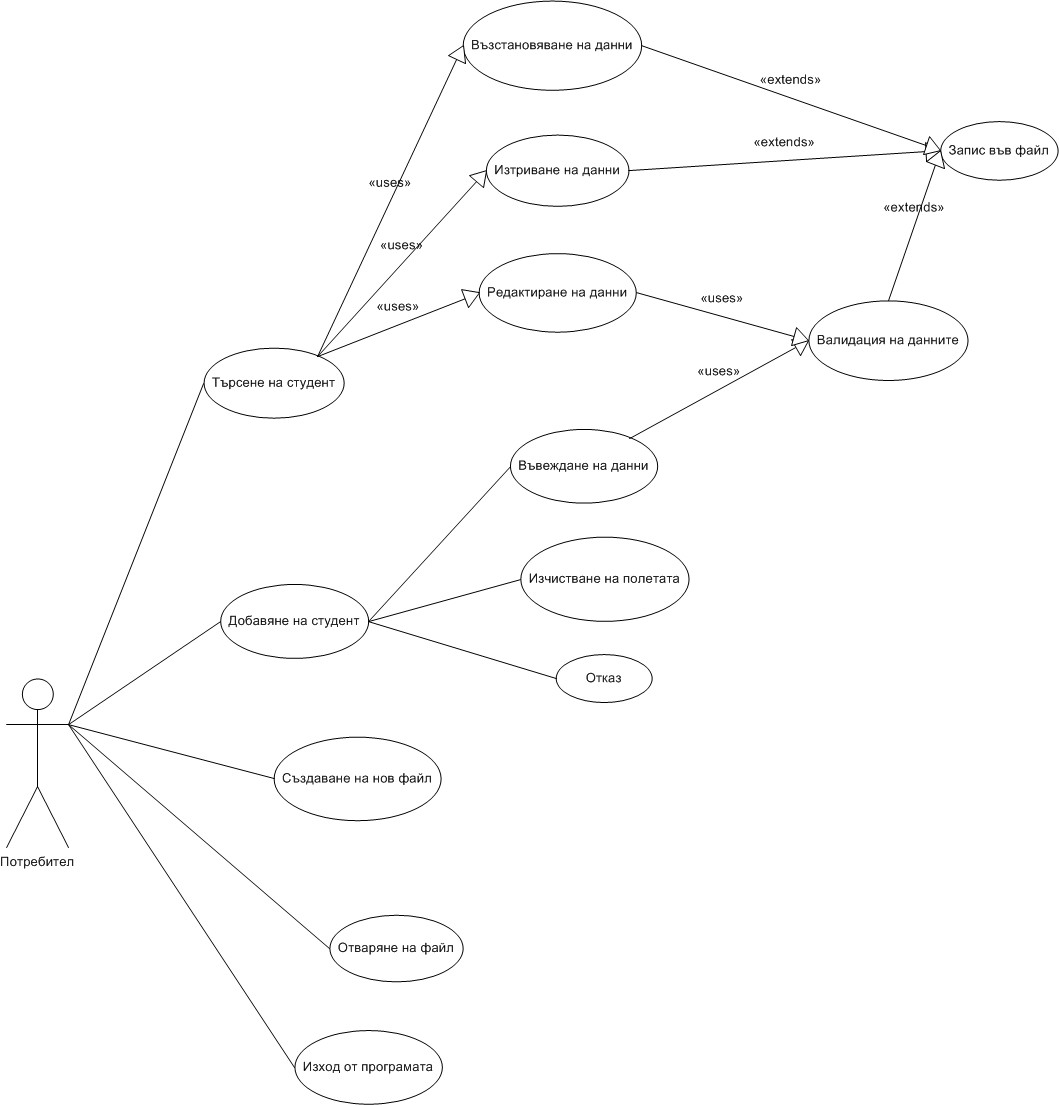
# Диаграми



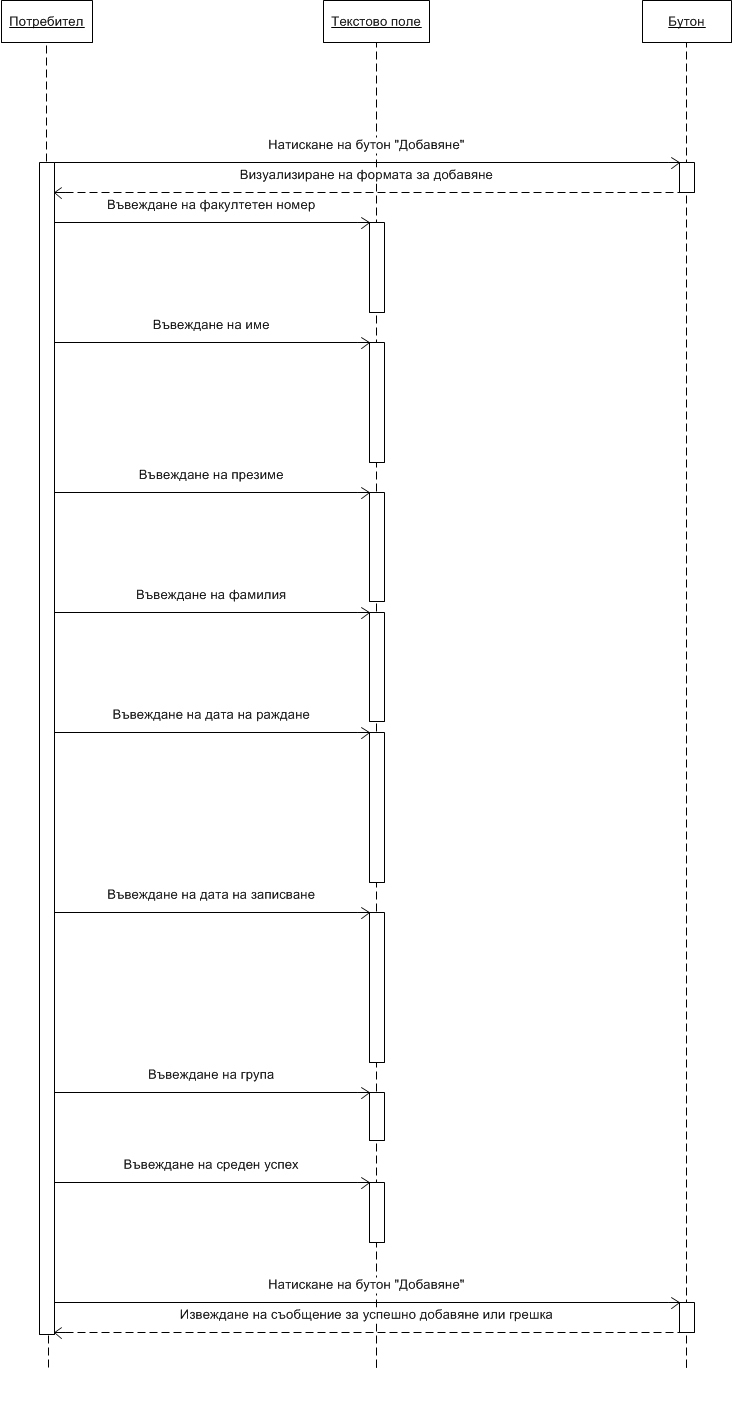
## Delphi диаграма



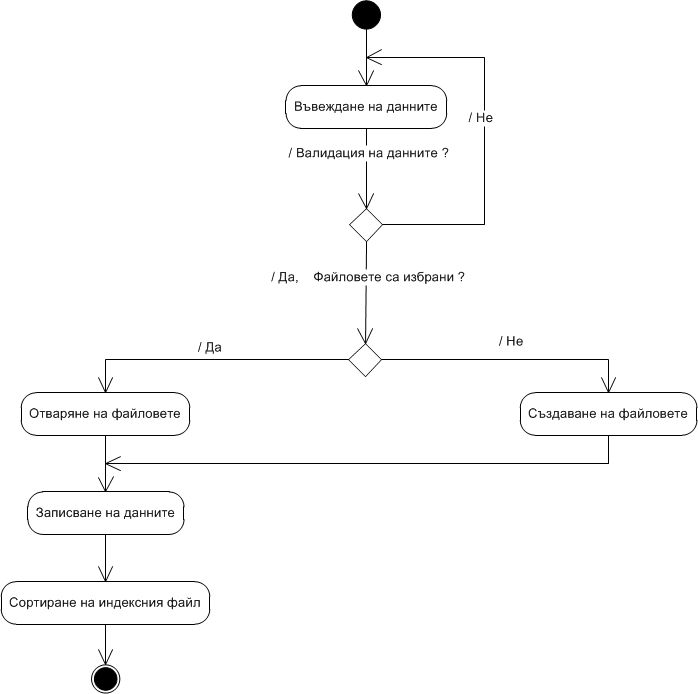
## Use Case



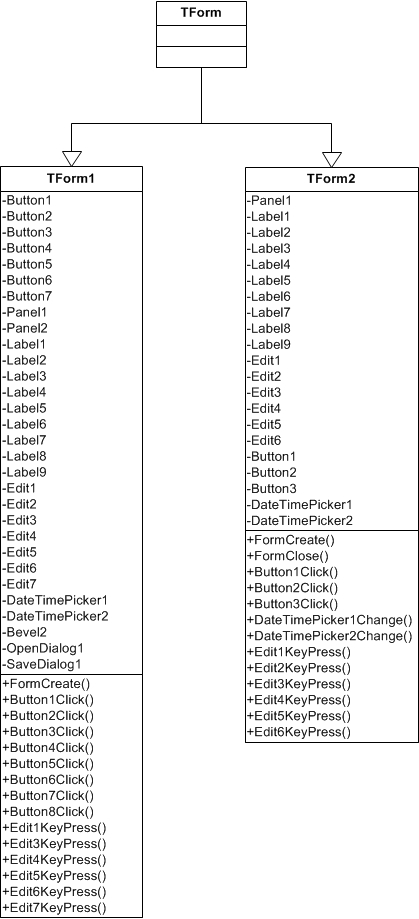
## Sequence



## Activity

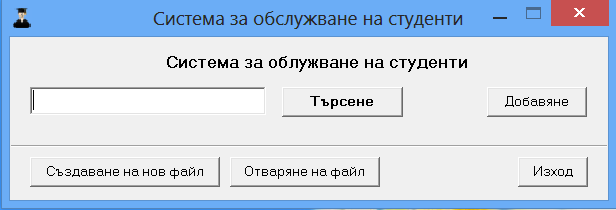


## Class



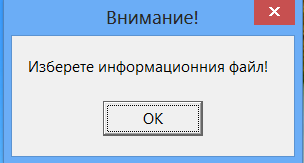
# Работа с програмата

При стартиране на програмния файл \*.ехе на екрана се визуализира прозорецът от фигура 1.



фиг. 1 Главен прозорец на програмата

Необходимо е да се избере (кликне върху) „Отваряне на файл“ или „Създаване на нов файл“. Ако се избере операцията „отваряне на файл“ се появява прозорец, в който се указва, че трябва да се избере информационния файл (фиг. 2).

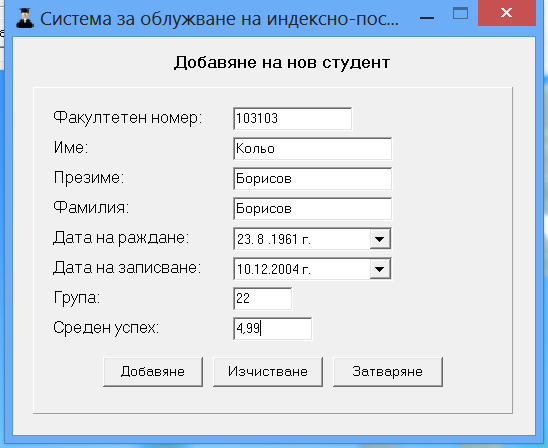


фиг. 2 Съобщение - избиране на информационния файл

След като се избере „ОК“ е необходимо да се избере информационния файл, а след това се появява друг прозорец със съобщение – „Изберете индексния файл“ и е необходимо да се избере индексния файл. След успешно избиране на двата файла програмата вече е готова за операции като търсене, редактиране, изтриване и т.н.

Ако ги няма, предварително създадени, тези файлове трябва или да се избере „Създаване на нов файл“, или - „Добавяне“. Ако се избере първият вариант се създават двата файла с указаните от нас имена в указаната от нас директория.

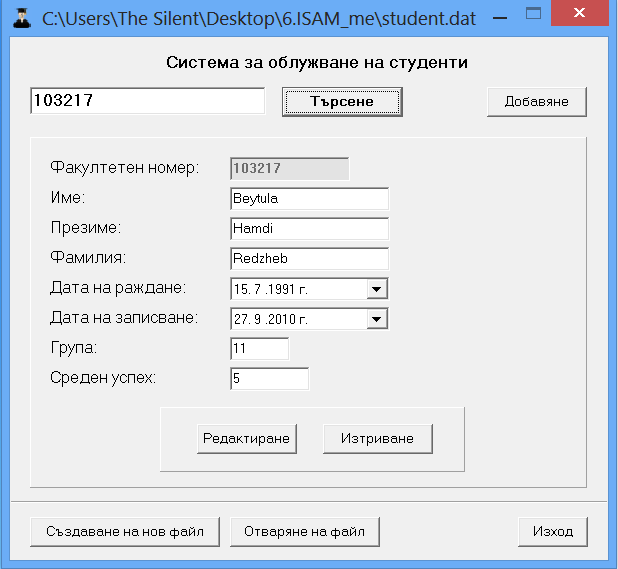
Ако се избере „Добавяне“ първо се попълват текстовите полета, а след това се избира „Добавяне“ от новопоявилият се прозорец – „Добавяне на нов студент“ (фиг. 3) и автоматично се появява друг прозорец за задаване името и мястото, където да се запише файла, при това информацията, попълнена в полетата се записва във файловете.



фиг. 3 Форма за добавяне на нов студент

Добавянето на нов студент във файл става по същия начин и когато файловете вече са избрани с тази разлика, че се появява само съобщение за успешно добавен студент. Ако се избере „Изчистване“, то цялата информация, записана в текстовите полета се изтрива. За затваряне на прозореца се избира „Затваряне“.

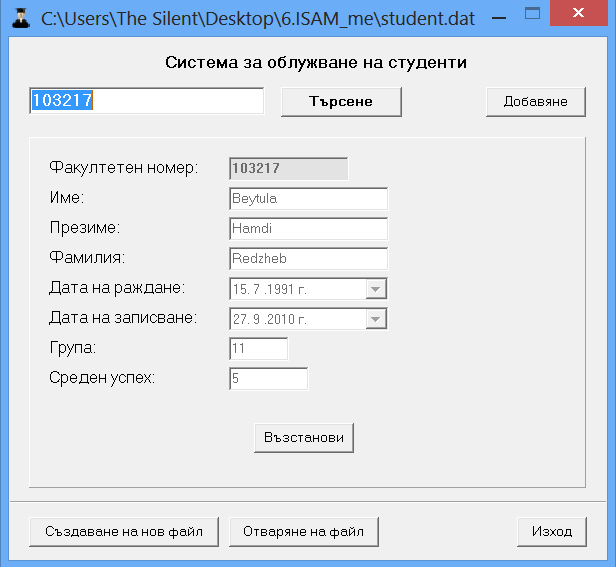
За търсене на студент във файл е необходимо да се напише факултетният номер на студента в текстовото поле на главната форма (прозорец) и да се натисне „Търсене“ (фиг. 4).



фиг. 4 Търсене на студент

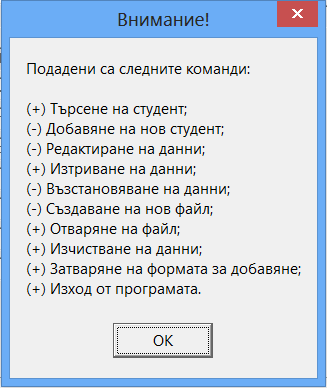
Ако факултетният номер е валиден, т.е съществува такъв студент във файла се визуализира цялата информация за този студент. Ако е необходимо да се направят корекции на данните, то трябва първо да се направят необходимите промени, а след това да се избере „Редактиране“.

При избор на „Изтриване“ данните за съответния студент не се изтриват „завинаги“. Програмата дава възможност за възстановяване на данните, примерно ако по невнимание е натиснат бутонът за изтриване или просто е необходимо това (фиг. 5).



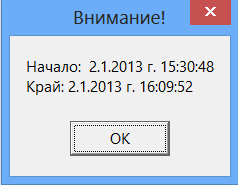
фиг. 5 Възстановяване на изтрити данни

За излизане от програмата се избира „Изход“. Появяват се два прозореца. В първия са изборени всички операции, които може да изпълнява програмата. Пред тези операции има или (-), или (+). (-) – показва, че операцията не е избрана, а (+) – обратното (фиг. 6).



фиг. 6 Използвани операции

Във втория прозорец се показва информацията за датата и времето за начало и край на сеанса(фиг. 7).



фиг. 7 Начало и край на сеанса

# Особености на програмата

* Размер – около 500 KB;
* ОС – Windows XP, 7, 8;
* Програмата е тествана с ОС Windows 7 Ultimate x64 и Windows 8 Professional x64 при резолюция на екрана 1920x1080 px.

# Код на програмата

## Unit1.pas

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ComCtrls, ExtCtrls;

type

TForm2 = class(TForm)

Panel1: TPanel;

Label1: TLabel;

Edit1: TEdit;

Label2: TLabel;

Edit2: TEdit;

Edit3: TEdit;

Label3: TLabel;

Label4: TLabel;

Edit4: TEdit;

Label5: TLabel;

DateTimePicker1: TDateTimePicker;

Label6: TLabel;

DateTimePicker2: TDateTimePicker;

Label7: TLabel;

Edit5: TEdit;

Label8: TLabel;

Edit6: TEdit;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label9: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure DateTimePicker1Change(Sender: TObject);

procedure DateTimePicker2Change(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormCreate(Sender: TObject);

procedure Edit6KeyPress(Sender: TObject; var Key: Char);

procedure Edit5KeyPress(Sender: TObject; var Key: Char);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure Edit2KeyPress(Sender: TObject; var Key: Char);

procedure Edit3KeyPress(Sender: TObject; var Key: Char);

procedure Edit4KeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit3, Unit2;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject);

begin

dob:='(+)';

if (Edit1.Text='') then

ShowMessage('Въведете факултетен номер!')

else if (length(Edit1.Text)<>6)then

ShowMessage('Въвели сте невалиден факултетен номер!')

else if (Edit2.Text='') then

ShowMessage('Въведете име!')

else if (Edit3.Text='') then

ShowMessage('Въведете презиме!')

else if (Edit4.Text='') then

ShowMessage('Въведете фамилия!')

else if (DateTimePicker1.Format=' ') then

ShowMessage('Въведете дата на раждане!')

else if (DateTimePicker2.Format=' ') then

ShowMessage('Въведете дата на записване!')

else if (Edit5.Text='') then

ShowMessage('Въведете група!')

else if (Edit6.Text='') then

ShowMessage('Въведете среден успех!')

else if ((StrToFloat(Edit6.Text)<2) or (StrToFloat(Edit6.Text)>6)) then

ShowMessage('Средният успех трябва да е между 2 и 6!')

else begin

student.fac\_num := Edit1.Text;

student.first\_name := Edit2.Text;

student.sec\_name := Edit3.Text;

student.last\_name := Edit4.Text;

student.date\_r := DateTimePicker1.Date;

student.date\_z := DateTimePicker2.Date;

student.group := StrToInt(Edit5.Text);

student.sr\_usp := StrToFloat(Edit6.Text);

student.deleted := false;

if (fname <> '') then begin

Reset(infF);

Seek(infF, FileSize(infF));

Write(infF, student);

Reset(indF);

Index.index := StrToInt(Edit1.Text);

Index.position := FileSize(infF);

CloseFile(infF);

Seek(indF, FileSize(indF));

write(indF, Index);

CloseFile(indF);

SortIndexFile();

ShowMessage('Студентът е добавен!');

Form2.Close;

end

else begin

opened := false;

ShowMessage('Запишете иформационния файл!');

if (Form1.SaveDialog1.Execute) then begin

opened := true;

fname := Form1.SaveDialog1.FileName;

AssignFile(infF, fname);

Rewrite(infF);

Write(infF, student);

Index.index := StrToInt(Edit1.Text);

Index.position := FileSize(infF);

CloseFile(infF);

end;

if (opened = true) then begin

ShowMessage('Запишете индексния файл!');

if (Form1.SaveDialog1.Execute) then begin

inFname := Form1.SaveDialog1.FileName;

AssignFile(indF, inFname);

Rewrite(indF);

Write(indF, Index);

CloseFile(indF);

ShowMessage('Студентът е добавен!');

Form2.Close;

syzd:='(+)';

Form1.Edit1.Text := '';

Form1.Button6.Top := 120;

Form1.Button7.Top := 120;

Form1.Button8.Top := 120;

Form1.Bevel2.Top := 100;

Form1.Height := 210;

Form1.Panel1.Visible := false;

end;

end;

end;

end;

end;

procedure TForm2.Button3Click(Sender: TObject);

begin

zatv:='(+)';

Form2.Close;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

izch:='(+)';

Edit1.Text := '';

Edit2.Text := '';

Edit3.Text := '';

Edit4.Text := '';

DateTimePicker1.Format := ' ';

DateTimePicker2.Format := ' ';

Edit5.Text := '';

Edit6.Text := '';

end;

procedure TForm2.DateTimePicker1Change(Sender: TObject);

begin

DateTimePicker1.Format := ShortDateFormat;

end;

procedure TForm2.DateTimePicker2Change(Sender: TObject);

begin

DateTimePicker2.Format := ShortDateFormat;

end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Edit1.Text := '';

Edit2.Text := '';

Edit3.Text := '';

Edit4.Text := '';

DateTimePicker1.Format := ' ';

DateTimePicker2.Format := ' ';

Edit5.Text := '';

Edit6.Text := '';

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

Edit1.Text := '';

Edit2.Text := '';

Edit3.Text := '';

Edit4.Text := '';

DateTimePicker1.Format := ' ';

DateTimePicker2.Format := ' ';

Edit5.Text := '';

Edit6.Text := '';

end;

procedure TForm2.Edit6KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9',#8,#44] = false then

Key := #10;

end;

procedure TForm2.Edit5KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9',#8] = false then

Key := #10;

end;

procedure TForm2.Edit1KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9',#8] = false then

Key := #10;

end;

procedure TForm2.Edit2KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9'] = true then

Key := #10;

end;

procedure TForm2.Edit3KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9'] = true then

Key := #10;

end;

procedure TForm2.Edit4KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9'] = true then

Key := #10;

end;

end.

## Unit2.pas

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, ComCtrls, Menus;

type

TForm1 = class(TForm)

Panel1: TPanel;

Label1: TLabel;

Edit1: TEdit;

Label2: TLabel;

Edit2: TEdit;

Edit3: TEdit;

Label3: TLabel;

Label4: TLabel;

Edit4: TEdit;

Label5: TLabel;

DateTimePicker1: TDateTimePicker;

Label6: TLabel;

DateTimePicker2: TDateTimePicker;

Label7: TLabel;

Edit5: TEdit;

Label8: TLabel;

Edit6: TEdit;

Edit7: TEdit;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Button8: TButton;

Label9: TLabel;

Bevel2: TBevel;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

Button4: TButton;

Panel2: TPanel;

procedure FormCreate(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure Edit6KeyPress(Sender: TObject; var Key: Char);

procedure Edit7KeyPress(Sender: TObject; var Key: Char);

procedure Edit3KeyPress(Sender: TObject; var Key: Char);

procedure Edit4KeyPress(Sender: TObject; var Key: Char);

procedure Edit5KeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

fname: string;

inFname: string;

opened: boolean;

startD: string;

stopD: string;

implementation

uses Unit1, Unit3;

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

Button6.Top := 120;

Button7.Top := 120;

Button8.Top := 120;

Bevel2.Top := 100;

Form1.Height := 210;

Panel1.Visible := false;

fname := '';

inFname := '';

startD := DateTimeToStr(Now);

tyrs:='(-)';

dob:='(-)';

red:='(-)';

iztr:='(-)';

vyzst:='(-)';

syzd:='(-)';

otv:='(-)';

izhod:='(-)';

izch:='(-)';

zatv:='(-)';

end;

procedure TForm1.Button5Click(Sender: TObject);

begin

Form2.Show;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

tyrs:='(+)';

if ((fname = '') or (inFname = '')) then

ShowMessage('Не е избран файл за четене!')

else begin

if Edit1.Text='' then

ShowMessage('Не сте въвели ключ/фак. номер за търсене')

else begin

Panel1.Visible := false;

positionFound := Search(StrToInt(Edit1.Text));

if (positionFound <> -1) then begin

Panel1.Visible := true;

Reset(infF);

Seek(infF,positionFound);

Read(infF,student);

CloseFile(infF);

Edit2.Text := student.fac\_num;

Edit3.Text := student.first\_name;

Edit4.Text := student.sec\_name;

Edit5.Text := student.last\_name;

DateTimePicker1.Date := student.date\_r;

DateTimePicker2.Date := student.date\_z;

Edit6.Text := IntToStr(student.group);

Edit7.Text := FloatToStr(student.sr\_usp);

Button6.Top := 480;

Button7.Top := 480;

Button8.Top := 480;

Bevel2.Top := 456;

Form1.Height :=570;

if student.deleted = true then begin

Button4.Visible := true;

Button4.Top := 286;

Button4.Left := 225;

Edit3.Enabled := false;

Edit4.Enabled := false;

Edit5.Enabled := false;

DateTimePicker1.Enabled := false;

DateTimePicker2.Enabled := false;

Edit6.Enabled := false;

Edit7.Enabled := false;

Panel2.Visible:= false;

end

else begin

Button4.Visible := false;

Edit3.Enabled := true;

Edit4.Enabled := true;

Edit5.Enabled := true;

DateTimePicker1.Enabled := true;

DateTimePicker2.Enabled := true;

Edit6.Enabled := true;

Edit7.Enabled := true;

Panel2.Visible:= true;

end;

end

else begin

Button6.Top := 120;

Button7.Top := 120;

Button8.Top := 120;

Bevel2.Top := 100;

Form1.Height := 210;

end;

end;

end;

end;

procedure TForm1.Button8Click(Sender: TObject);

begin

izhod:='(+)';

ShowMessage('Подадени са следните команди:' + #13#10 +

#13#10 + tyrs + ' Търсене на студент;' +

#13#10 + dob + ' Добавяне на нов студент;' +

#13#10 + red + ' Редактиране на данни;' +

#13#10 + iztr + ' Изтриване на данни;' +

#13#10 + vyzst + ' Възстановяване на данни;' +

#13#10 + syzd + ' Създаване на нов файл;' +

#13#10 + otv + ' Отваряне на файл;' +

#13#10 + izch + ' Изчистване на данни;' +

#13#10 + zatv + ' Затваряне на формата за добавяне;' +

#13#10 + izhod + ' Изход от програмата.');

stopD := DateTimeToStr(Now);

ShowMessage('Начало: ' + startD + #13#10 + 'Край: ' + stopD);

Form1.Close;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

syzd:='(+)';

opened := false;

ShowMessage('Запишете иформационния файл!');

if (Form1.SaveDialog1.Execute) then begin

fname := Form1.SaveDialog1.FileName;

Form1.Caption := fname;

AssignFile(infF, fname);

Rewrite(infF);

opened := true;

CloseFile(infF);

end;

if (opened = true) then begin

ShowMessage('Запишете индексния файл!');

if (Form1.SaveDialog1.Execute) then begin

inFname := Form1.SaveDialog1.FileName;

AssignFile(indF, inFname);

Rewrite(indF);

CloseFile(indF);

Edit1.Text := '';

Button6.Top := 120;

Button7.Top := 120;

Button8.Top := 120;

Bevel2.Top := 100;

Form1.Height := 210;

Panel1.Visible := false;

end;

end;

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

otv:='(+)';

opened := false;

ShowMessage('Изберете информационния файл!');

if (Form1.OpenDialog1.Execute) then begin

fname := Form1.OpenDialog1.FileName;

Form1.Caption := fname;

AssignFile(infF, fname);

opened := true;

end;

if (opened = true) then begin

ShowMessage('Изберете индексния файл!');

if (Form1.OpenDialog1.Execute) then begin

inFname := Form1.OpenDialog1.FileName;

AssignFile(indF, inFname);

Edit1.Text := '';

Button6.Top := 120;

Button7.Top := 120;

Button8.Top := 120;

Bevel2.Top := 100;

Form1.Height := 210;

Panel1.Visible := false;

end;

end;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

iztr:='(+)';

if student.deleted = false then begin

Button4.Visible := false;

Reset(infF,fname);

Seek(infF, positionFound);

student.deleted := true;

write(infF, student);

ShowMessage('Данните за студент с факултетен номер: '+student.fac\_num+' са изтрити!');

CloseFile(infF);

Button4.Visible := true;

Button4.Top := 286;

Button4.Left := 225;

Edit3.Enabled := false;

Edit4.Enabled := false;

Edit5.Enabled := false;

DateTimePicker1.Enabled := false;

DateTimePicker2.Enabled := false;

Edit6.Enabled := false;

Edit7.Enabled := false;

Panel2.Visible:= false;

end;

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

vyzst:='(+)';

if student.deleted = true then begin

Reset(infF,fname);

Seek(infF, positionFound);

student.deleted := false;

write(infF, student);

ShowMessage('Данните за студент с факултетен номер: '+student.fac\_num+' са възстановени!');

CloseFile(infF);

Button4.Visible := false;

Edit3.Enabled := true;

Edit4.Enabled := true;

Edit5.Enabled := true;

DateTimePicker1.Enabled := true;

DateTimePicker2.Enabled := true;

Edit6.Enabled := true;

Edit7.Enabled := true;

Panel2.Visible:= true;

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

red:= '(+)';

if (Edit3.Text='') then

ShowMessage('Въведете име!')

else if (Edit4.Text='') then

ShowMessage('Въведете презиме!')

else if (Edit5.Text='') then

ShowMessage('Въведете фамилия!')

else if (DateTimePicker1.Format=' ') then

ShowMessage('Въведете дата на раждане!')

else if (DateTimePicker2.Format=' ') then

ShowMessage('Въведете дата на записване!')

else if (Edit6.Text='') then

ShowMessage('Въведете група!')

else if (Edit7.Text='') then

ShowMessage('Въведете среден успех!')

else if ((StrToFloat(Edit7.Text)<2) or (StrToFloat(Edit7.Text)>6)) then

ShowMessage('Средният успех трябва да е между 2 и 6!')

else if student.deleted = false then begin

Reset(infF);

Seek(infF, positionFound);

student.first\_name := Edit3.Text;

student.sec\_name := Edit4.Text;

student.last\_name := Edit5.Text;

student.date\_r := DateTimePicker1.Date;

student.date\_z := DateTimePicker2.Date;

student.group := StrToInt(Edit6.Text);

student.sr\_usp := StrToFloat(Edit7.Text);

student.deleted := false;

write(infF, student);

ShowMessage('Данните за студент с факултетен номер: '+student.fac\_num+' са променени!');

CloseFile(infF);

end;

end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9',#8,#13] = false then

Key := #10;

if Key = #13 then

Button1.OnClick(Sender);

end;

procedure TForm1.Edit6KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9',#8] = false then

Key := #10;

end;

procedure TForm1.Edit7KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9',#8,#44] = false then

Key := #10;

end;

procedure TForm1.Edit3KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9'] = true then

Key := #10;

end;

procedure TForm1.Edit4KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9'] = true then

Key := #10;

end;

procedure TForm1.Edit5KeyPress(Sender: TObject; var Key: Char);

begin

if Key in ['0'..'9'] = true then

Key := #10;

end;

end.

## Unit3.pas

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ComCtrls;

type

TIndex = Record

index : integer;

position : integer;

end;

IndFile = File of TIndex;

TStudent = Record

fac\_num: string[6];

first\_name: string[25];

sec\_name: string[25];

last\_name: string[25];

group: integer;

sr\_usp: real;

date\_r: TDate;

date\_z: TDate;

deleted: boolean;

end;

InfFile = File of TStudent;

var

indF: IndFile;

infF: InfFile;

student: TStudent;

index: TIndex;

positionFound: integer;

{\\ Podadeni Komandi}

tyrs:string;

dob:string;

red:string;

iztr:string;

vyzst:string;

syzd:string;

otv:string;

izhod:string;

izch:string;

zatv:string;

{// Podadeni Komandi}

procedure SortIndexFile();

function Search(keyword: integer): integer;

implementation

procedure SortIndexFile();

var

M,J,H : Integer;

Buffer : TIndex;

IndexArray : array[1..100] of TIndex;

begin

Reset(IndF);

for J := 1 to FileSize(IndF) do

Read(IndF,IndexArray[J]);

H := FileSize(IndF) Div 2;

while H > 0 do begin

M := H + 1;

repeat

J := M - H;

while J > 0 do begin

if IndexArray[J].index <= IndexArray[J+H].index then

J := 0

else begin

Buffer := IndexArray[J];

IndexArray[J] := IndexArray[J + H];

IndexArray[J + H] := Buffer;

end;

J := J - H;

end;

M := M + 1;

until M > FileSize(IndF);

H := H div 2;

end;

Seek(IndF, 0);

for J := 1 to FileSize(IndF) do

Write(IndF,IndexArray[J]);

CloseFile(IndF);

end;

function Search(keyword: integer): integer;

var

I,L,D:Integer;

begin

Reset(indF);

if FileSize(indF) = 0 then begin

Result := -1;

exit;

end;

{cycles := 0;}

L := 0; D := FileSize(indF)-1;

repeat

{cycles := cycles + 1; }

I := (L + D) div 2;

Seek(indF, I);

Read(indF, Index);

if keyword > Index.index then

L := I + 1

else if keyword < Index.index then

D := I - 1;

until (keyword = Index.index) or (L > D);

if (keyword = Index.index) then

Result := Index.position-1

else

Result := -1;

end;

end.